Processing Large Amounts of Images on Hadoop with OpenCV

Timofei Epanchintsev[1,2] and Andrey Sozykin[1,2]

[1] IMM UB RAS, Yekaterinburg, Russia,
[2] Ural Federal University, Yekaterinburg, Russia
{eti,avs}@imm.uran.ru

Abstract. Modern image collections cannot be processed efficiently on one computer due to large collection sizes and high computational costs of modern image processing algorithms. Hence, image processing often requires distributed computing. However, distributed computing is a complicated subject that demands deep technical knowledge and often cannot be used by researches who develop image processing algorithms. The framework is needed that allows the researches to concentrate on image processing tasks and hides from them the complicated details of distributed computing. In addition, the framework should provide the researches with the familiar image processing tools. The paper describes the extension to the MapReduce Image Processing (MIPr) framework that provides the ability to use OpenCV in Hadoop cluster for distributed image processing. The modified MIPr framework allows the development of image processing programs in Java using the OpenCV Java binding. The performance testing of created system on the cloud cluster demonstrated near-linear scalability.

Keywords: OpenCV · Hadoop · MapReduce · image processing · distributed processing

1 Introduction

Image processing is one of the important tasks in many areas of research such as medical imaging, remote sensing, astronomy, Internet analysis, etc. The industry also widely uses image processing.

Nowadays, image processing often requires distributed computing. The sizes of modern image collections, for example, the Yahoo 100 million creative commons flickr images for research [6], are large (terabytes and petabytes of data); such collections cannot be stored and processed efficiently on a single computer. In addition, contemporary image processing algorithms are becoming very complex and, hence, computationally intensive.

However, distributed computing is a complicated subject that requires deep technical knowledge. Many scientists, who developed image processing algorithms, does not have the qualification required to use distributed computing efficiently. They need an image processing framework that hides the complicated

details of distributed computing from the application developers and allows them to concentrate on image processing tasks and algorithms.

Previously, we have developed the MapReduce Image Processing framework (MIPr) [4], which allows to process images on Apache Hadoop cluster using the MapReduce technology. MIPr provides the image representations in the Hadoop internal format, which are suitable for MapReduce processing, and the input/output tools for image processing integration into the Hadoop data workflow. In addition, MIPr includes high-level image processing API for application developers who are not familiar with Hadoop.

However, providing the application developers with a convenient framework is not enough for efficient distributed image processing. The developers must be able to use image processing tools that they already know together with the framework. In such case, the developers can quickly create software for distributed computing framework, or even use their existing software. Otherwise, the developers will have to implement image processing algorithms from scratch, which require a lot of time and labor.

In this paper we describe an approach for using the popular open source computer vision library OpenCV [1] for distributed image processing on a Hadoop cluster with the help of the MIPr framework. OpenCV includes more than 2500 computer vision and machine learning algorithms for image and video processing.

2 Related Work

Several systems can be used for image processing in Hadoop. HIPI (Hadoop Image Processing Interface) [5] is a framework specifically designed to enable image processing in Hadoop. OpenIMAJ (Open Intelligent Multimedia Analysis for Java) [2] contains several Hadoop implementation of computer vision algorithms. In contrast to our work neither system supports the OpenCV library.

The image processing cloud with the OpenCV support was built at the Prairie View A&M University (PVAMU)[7]. The PVAMU cloud provides the Platform as a Service for distributed image processing and storage, which includes the Hadoop cluster and the OpenCV library. Two types of Hadoop InputFormat and OutputFormat based on OpenCV have been developed: for image and video files. Hadoop Streaming is used for data processing. Hence, it is possible to use various programming languages to develop image processing algorithms. Unfortunately, the authors did not share their implementation of OpenCV support for Hadoop. Hence, the system can be used only in the PVAMU cloud, which is not very convenient. Sometimes, copying a large image collection to the remote cloud through the network requires more time than the processing of the images. Our MIPr framework is open sourced and freely available to use in any Hadoop cluster, not only at the cloud platform. On the other hand, the PVAMU cloud has two advantages compared with the MIPr: the ability to process not only images, but also videos, and the support of multiple languages for implementation of image and video processing algorithms.

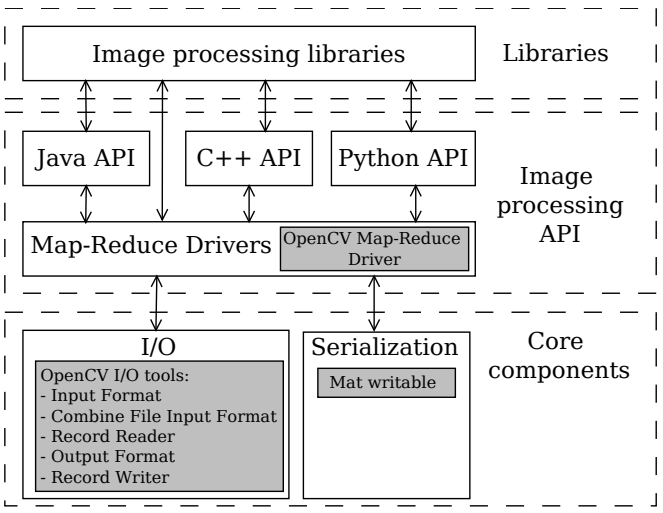

Fig. 1. Extended architecture of the MIPr framework (new blocks are gray)

3 Integration of OpenCV Library into the MIPr Framework

The MIPr framework has an open architecture, which allows adding new image representation based on various libraries. The extended architecture of MIPr with the support of OpenCV library is presented in Fig 1. The extensions include an OpenCV-based image representation in the Hadoop internal format, input/output tools for inclusion of OpenCV-based images into the Hadoop data workflow, and MapReduce Driver to execute Hadoop programs with the OpenCV support.

3.1 Image Representation

OpenCV uses the Mat class as an image container, and most of OpenCV image processing functions accept Mat as an argument. Hence, the Mat class was chosen to build the OpenCV-based image format in MIPr. Similar to other MIPr image representation formats, the Hadoop Writable interface was used for serialization. By combining these two components, the *MatWritable* image format was created, which is suitable for use as a value in MapReduce programs.

3.2 Input/Output tools

For reading images from the Hadoop Distributed File System (HDFS) into the memory for processing, two types of InputFormat for OpenCV have been developed: *OpenCVFileInputFormat* and *OpenCVCombineFileInputFormat*. *OpenCVFileInputFormat* reads one image at a time and is suitable for

big images (larger than the HDFS block size). In contrast with it, *OpenCVCombineFileInputFormat* reads several images from the same HDFS block in one operation and is suitable for processing large amount of small files. Both InputFormats produce pairs <file name, MatWritable> (*OpenCVCombineFileInputFormat* can produce several pairs if the HDSF block contains several images). To actually read an image from HDFS and present it in the *MatWritable* format, the *OpenCVRecordReader* was developed, which is used both by the *OpenCVFileInputFormat* and *OpenCVCombineFileInputFormat*.

For writing the OpenCV-based images to HDFS after processing, special implementations of OutputFormat and RecordWriter, which are capable of working with *MatWritable*, were developed. In contrast to the traditional Hadoop approach, the created components preserve the original names of image files.

3.3 MapReduce Driver

MapReduce driver is intended to execute the MapReduce program on the cluster. The developed OpenCV MapReduce driver sets up the appropriate InputFormat and OutputFormat, defines the MapReduce values type as a *MatWritable*, loads OpenCV native C++ library, and executes the job on the Hadoop cluster. To distribute the binary OpenCV C++ library to the cluster nodes, the Hadoop distributed cache is used.

3.4 Image Processing

The MIPr extensions use the OpenCV Java binding. In contrast to the PVAMU cloud, MIPr utilizes the standard Hadoop Mappers and Reducers for image processing. Hence, only Java is supported for developing image processing algorithms in MIPr using OpenCV.

4 Performance evaluation

In order to estimate the performance and scalability of the proposed approach, two computational experiments with the modified MIPr implementation have been conducted. The Microsoft Azure HDInsight cluster in the Microsoft Azure cloud platform have been used for the experiments. The cluster consists of 18 nodes: 2 head nodes and 16 computational nodes. The configuration of the nodes is presented in the Table 1.

The MIRFLICKR-1M image collection (one million of images downloaded from Flickr, 118GB total size) [3] was used as a benchmark dataset. The following image processing operations were performed during the experiments:

- conversion of color images to grayscale;
- face detection;
- Canny edge detection.

Table 1. Configuration of the Hadoop cluster nodes

Configuration parameter	Value
CPU	Intel Xeon E5-2660 @ 2.20GHz, 4 cores
RAM	14 GB
Local HDD	200GB SSD
Network interface	Gigabit Ethernet
Operating System	Linux Ubuntu 12.04 LTS

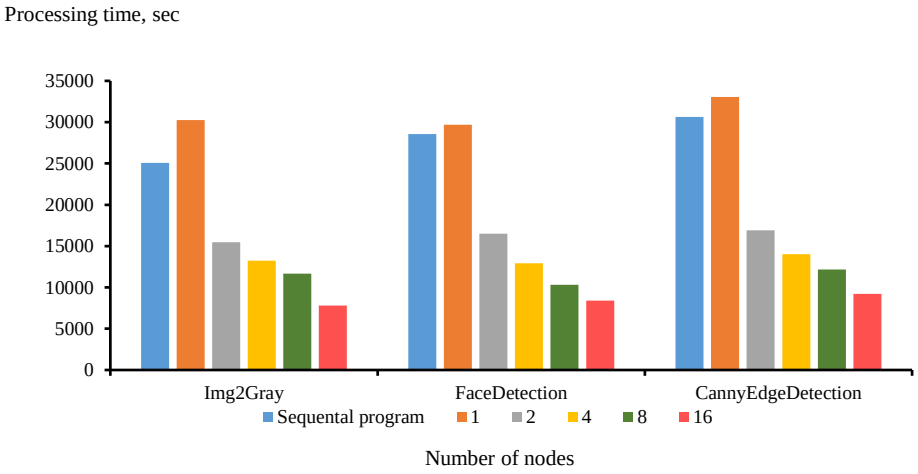

Fig. 2. Image processing time depending on the number of nodes in the cluster

The purpose of the first experiment was to estimate the scalability of the modified MIPr implementation with the growing number of nodes in the cluster. During this experiment, the entire MIRFLICKR-1M image collection was processed using varying number of nodes. In addition to the cluster implementation, the performance of sequential image processing was measured. The results are presented in Fig. 2. With the exception of single node cluster, the system has near-linear scalability. A poor performance of the single node cluster, as compared with the sequential implementation, is caused by the Hadoop overhead.

The second experiment was aimed at estimating how the system scales with the increasing volume of images. For this purpose, the whole cluster was used to process varying number of images. The results are presented in Fig. 3 (logarithmic scale). For small datasets (the number of images is less than 10000), there is no difference in processing time due to the cluster overhead. However, the scalability for large image collection processing is near-linear.

Near-linear scalability of the modified MIPr implementation is an expected result, because images are processed independently.

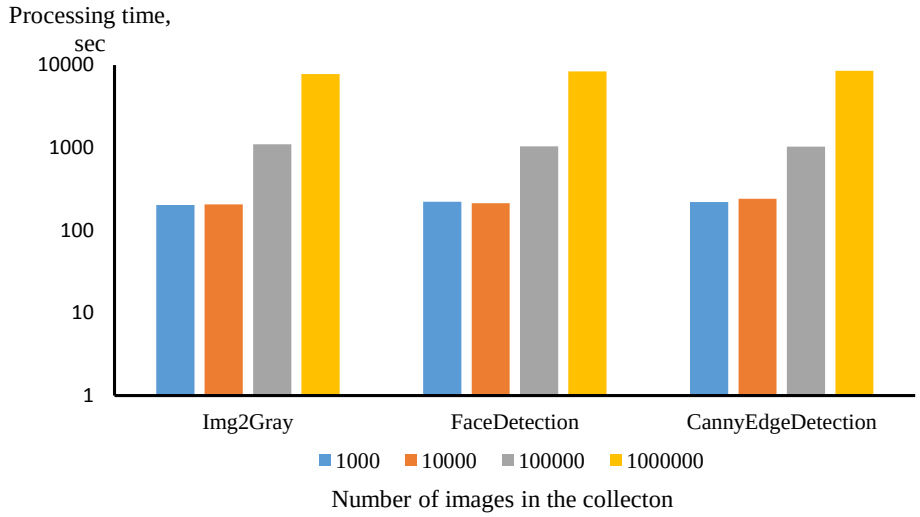

Fig. 3. Image processing time depending on the number of images in the collection (logarithmic scale)

5 Conclusion

The approach to use OpenCV library for distributed image processing on a Hadoop cluster was presented. The approach is based on the previously developed MIPr framework for image processing in Hadoop. The MIPr framework was extended to include OpenCV-based image representation in the Hadoop internal format, tools for reading images from HDFS and writing processed images back to HDFS, and drivers for MapReduce jobs with OpenCV support. The extension allows to develop image processing programs in Java using OpenCV Java bindings.

The performance of the modified MIPr implementation was evaluated. For large clusters and image datasets the scalability is near-linear. Poor performance of single node cluster and small dataset processing is caused by Hadoop overhead.

MIPr framework is open sourced and available for free at `https://github.com/sozykin/mipr`.

Future work include porting MIPr to Apache Spark [8] for in-memory image processing and providing the ability to use Python and C++ for developing MapReduce image processing programs.

Acknowledgements. The reported study was supported by RFBR (research project No 14-07-31324 mol_a) and by Microsoft Azure for Research Russia Initiative. Our work was performed using the “Uran” supercomputer from Institute of Mathematics and Mechanics UrB RAS.

References

1. Bradski, G.: The opencv library. Dr. Dobb's Journal of Software Tools (2000)
2. Hare, J.S., Samangooei, S., Dupplaw, D.P.: Openimaj and imageterrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In: Proceedings of the 19th ACM international conference on Multimedia. pp. 691–694. MM '11, ACM, New York, NY, USA (2011), http://doi.acm.org/10.1145/2072298.2072421
3. Huiskes, M.J., Thomee, B., Lew, M.S.: New trends and ideas in visual concept detection: The mir flickr retrieval evaluation initiative. In: Proceedings of the International Conference on Multimedia Information Retrieval. pp. 527–536. MIR '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1743384.1743475
4. Sozykin, A., Epanchintsev, T.: Mipr – a framework for distributed image processing using hadoop. In: Application of Information and Communication Technologies (AICT), 2015 IEEE 9th International Conference on. pp. 35–39 (2015)
5. Sweeney, C., Liu, L., Arietta, S., Lawrence, J.: HIPI: A Hadoop Image Processing Interface for Image-based MapReduce Tasks. B.s. thesis, University of Virginia (2011)
6. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: The new data and new challenges in multimedia research. arXiv preprint arXiv:1503.01817 (2015)
7. Yan, Y., Huang, L.: Large-scale image processing research cloud. In: CLOUD COMPUTING 2014 : The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization. pp. 88–93 (2014)
8. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: Cluster computing with working sets. In: Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing. pp. 10–10. HotCloud'10, USENIX Association, Berkeley, CA, USA (2010), http://dl.acm.org/citation.cfm?id=1863103.1863113